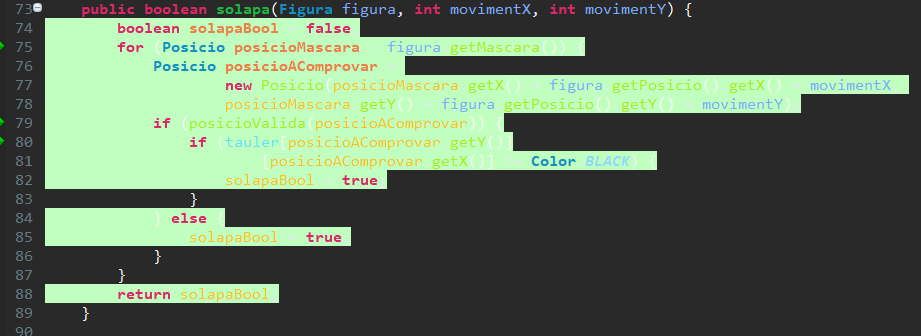
**Proves de caixa blanca**

Hem utilitzat les tècniques de caixa blanca en les parts del codi on teniem funcions amb una certa complexitat i podiem analitzar l’algoritme més detingudament. Les funcions que hem considerat testejar son les de solapa(),comprovarFilesCompletades() i eliminarFilaCompletada().

Hem controlat el control flow amb Eclemma:

**1. Classe Tauler**

**Funció solapa:**



-*Statement coverage:* Totes les sentències s’estan executant com a mínim un cop.

-*Decision coverage:* Totes les decisions prenen els valors true i false.

-*Condition coverage:* Com en aquest codi no tenim cap condició amb més d’una decisió, el fet de que es compleixi el decision coverage ja ens garantitza que s’està complint el condition coverage.

-*Loop testing:* Hem realitzat loop testing per al bucle for de la línea 74. Com que aquest bucle depén del nombre de posicions, i el Tetris només té figures amb 1, 2 o 4 rotacions, només podem probar el bucle per aquests tres casos.

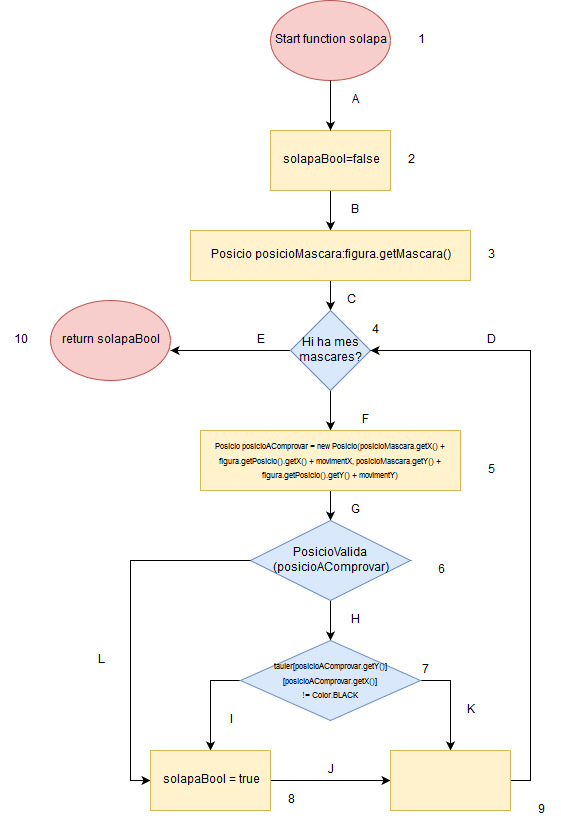
*-Path coverage:* Les proves realitzades compleixen tots els paths independents de la funció.

Complexitat ciclomàtica:

M = num. de condicions + 1 = 3 + 1 = 4

O també:

M = num.arestes - num.nodes + 2 = 12 - 10 + 2 = 4



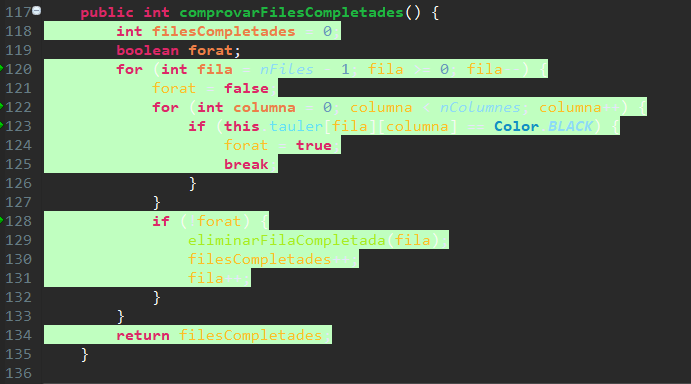
1. (1A, 2B, 3C, 4E, 10)

2. (1A, 2B, 3C, 4F, 5G, 6H, 7I, 8J, 9D, 4E, 10)

3. (1A, 2B, 3C, 4F, 5G, 6L, 8J, 9D, 4E, 10)

4. (1A, 2B, 3C, 4F, 5G, 6H, 7K, 9D, 4E, 10)

**Funció comprovarFilesCompletades():**



-*Statement coverage:* Totes les sentències s’estan executant com a mínim un cop.

-*Decision coverage:* Totes les decisions prenen els valors true i false.

-*Condition coverage:* Com en aquest codi no tenim cap condició amb més d’una decisió, el fet de que es compleixi el decision coverage ja ens garantitza que s’està complint el condition coverage.

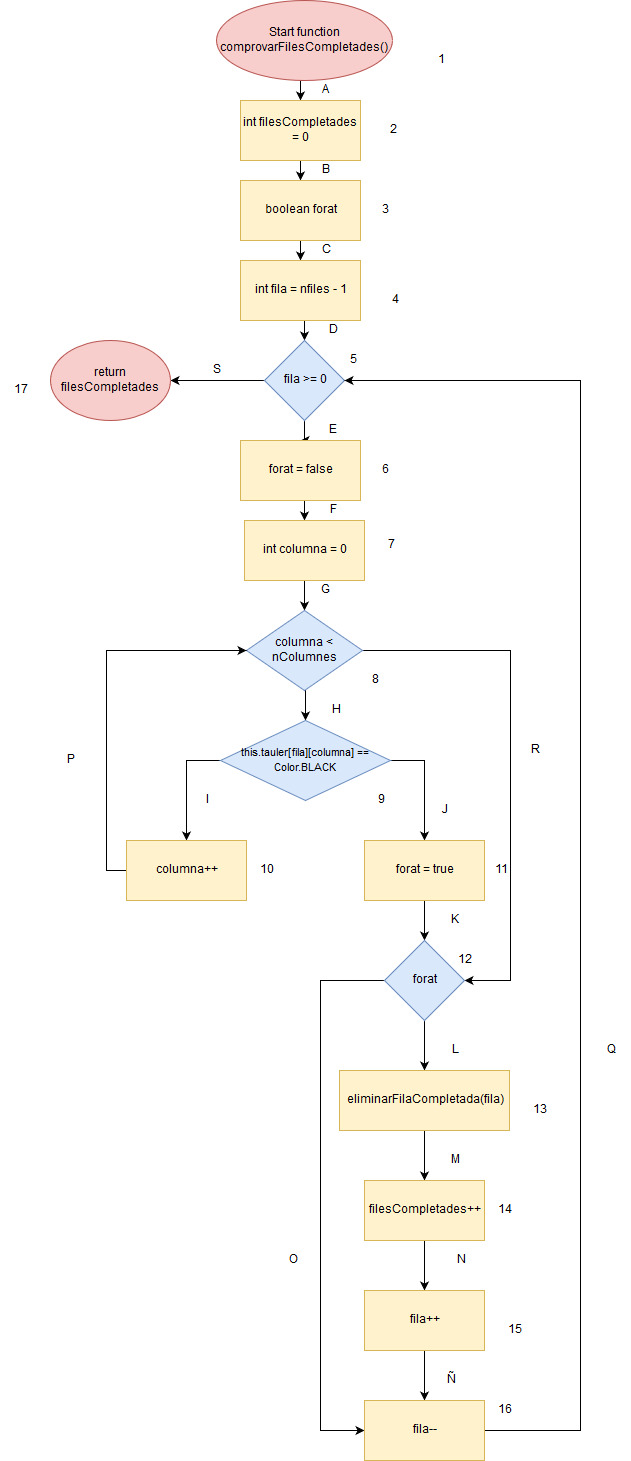
*-Path coverage:* Les proves realitzades compleixen tots els paths independents de la funció.

Complexitat ciclomàtica:

M = num. de condicions + 1 = 4 + 1 = 5

O també:

M num.arestes - num.nodes + 2 = 20 - 17 + 2 = 3 + 2 = 5



1. (1A, 2B, 3C, 4D, 5S, 17)

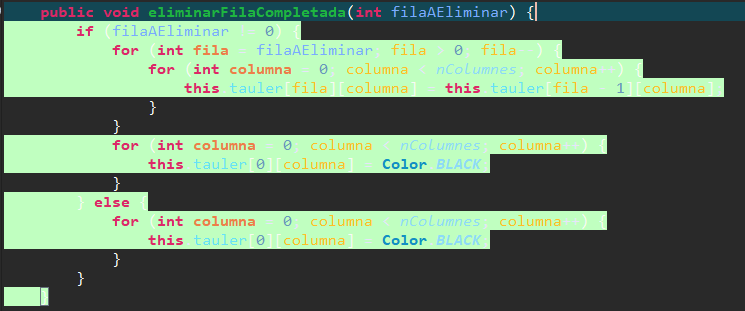
2. (1A, 2B, 3C, 4D, 5E, 6F, 7G, 8H, 9J, 11K, 12L, 13M, 14N, 15Ñ, 16Q, 5S, 17)

3. (1A, 2B, 3C, 4D, 5E, 6F, 7G, 8R, 12L, 13M, 14N, 15Ñ, 16Q, 5S, 17)

4. (1A, 2B, 3C, 4D, 5E, 6F, 7G, 8H, 9J, 11K, 12O, 16Q, 5S, 17)

5. (1A, 2B, 3C, 4D, 5E, 6F, 7G, 8H, 9I, 10P, 8H, 9J, 11K, 120, 16Q, 5S, 17)

**Funció eliminarFilaCompletada():**



-*Statement coverage:* Totes les sentències s’estan executant com a mínim un cop.

-*Decision coverage:* Totes les decisions prenen els valors true i false.

-*Condition coverage:* Com en aquest codi no tenim cap condició amb més d’una decisió, el fet de que es compleixi el decision coverage ja ens garantitza que s’està complint el condition coverage.

-*Loop testing:* Testejarem el primer bucle de tots (que depén de filaAEliminar). Primer no entrarem al bucle, després farem una passada, dues, 10 passades i nFiles – 1 passades.

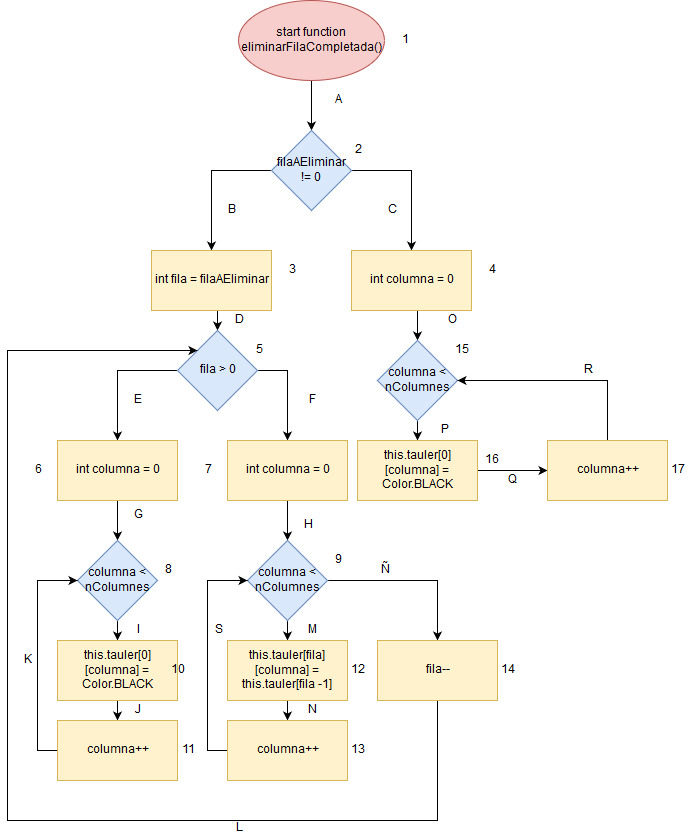
*-Path coverage:* Les proves realitzades compleixen tots els paths independents de la funció.

Complexitat ciclomàtica:

M = num. de condicions + 1 = 5 + 1 = 6

O també:

M num.arestes - num.nodes + 2 = 21 - 17 + 2 = 4 + 2 = 6



1. (1A, 2C, 4O, 15P, 16Q, 17R)

2. (1A, 2B, 3D, 5F, 7H)

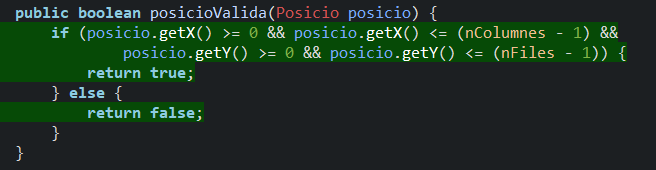
3. (1A, 2B, 3D, 5E, 6G)

4. (1A, 2B, 3D, 5E, 6G, 8I, 10J, 11K, 8)

5. (1A, 2B, 3D, 5F, 7H, 9M, 12N, 13

6. (1A, 2B, 3D, 5F, 7H, 9Ñ, 14L, 5F, 7H, 9M, 12N, 13S, 9)

**Funció posicioValida():**



-*Statement coverage:* Totes les sentències s’estan executant com a mínim un cop.

-*Decision coverage:* Totes les decisions prenen els valors true i false.

-*Condition coverage:* Com en aquest codi no tenim cap condició amb més d’una decisió, el fet de que es compleixi el decision coverage ja ens garantitza que s’està complint el condition coverage.

*-Path coverage:* Les proves realitzades compleixen tots els paths independents de la funció.

Complexitat ciclomàtica:

M = num. de condicions + 1 = 1 + 1 = 2

O també:

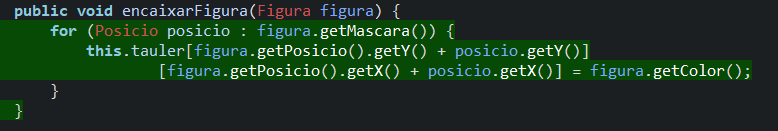
M num.arestes - num.nodes + 2 = 3 - 3 + 2 = 0 + 2 = 2

C:\Users\marcm\AppData\Local\Microsoft\Windows\INetCache\Content.Word\posicioValida.png

1. (1A, 2-Si, 3)

2. (1A, 2-No, 4)

**Funció encaixarFigura():**



-*Statement coverage:* Totes les sentències s’estan executant com a mínim un cop.

-*Decision coverage:* Com que en aquest codi no hi ha cap condició, el decision coverage no es pot complir.

-*Condition coverage:* Com en aquest codi no hi ha cap condició, el decision coverage no es pot complir.

-*Loop testing:* Com que el bucle de la funció depén del nombre de posicions, i el Tetris només té figures amb 1, 2 o 4 rotacions, només podem probar el bucle per aquests tres casos.

*-Path coverage:* Les proves realitzades compleixen tots els paths independents de la funció.

Complexitat ciclomàtica:

M = num. de condicions + 1 = 1 + 1 = 2

O també:

M num.arestes - num.nodes + 2 = 4 - 4 + 2 = 0 + 2 = 2

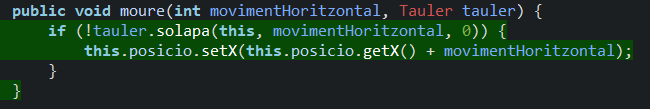
C:\Users\marcm\AppData\Local\Microsoft\Windows\INetCache\Content.Word\posicioValida.png

1. (1A, 2-No, 4 )

2. (1A, 2-Si, 3-B, 2-No, 4)

**2. Classe Figura**

**Funció moure():**



-*Statement coverage:* Totes les sentències s’estan executant com a mínim un cop.

-*Decision coverage:* Totes les decisions prenen els valors true i false.

-*Condition coverage:* Com en aquest codi no tenim cap condició amb més d’una decisió, el fet de que es compleixi el decision coverage ja ens garantitza que s’està complint el condition coverage.

*-Path coverage:* Les proves realitzades compleixen tots els paths independents de la funció.

Complexitat ciclomàtica:

M = num. de condicions + 1 = 1 + 1 = 2

O també:

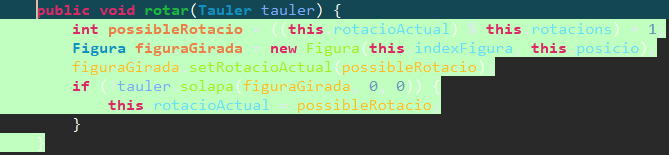
M num.arestes - num.nodes + 2 = 4 - 4 + 2 = 0 + 2 = 2

C:\Users\marcm\AppData\Local\Microsoft\Windows\INetCache\Content.Word\posicioValida(1).png

1. (1A, 2-No, 4 )

2. (1A, 2-Si, 3-B, 4)

**Funció rotar():**



-*Statement coverage:* Totes les sentències s’estan executant com a mínim un cop.

-*Decision coverage:* Totes les decisions prenen els valors true i false.

-*Condition coverage:* Com en aquest codi no tenim cap condició amb més d’una decisió, el fet de que es compleixi el decision coverage ja ens garantitza que s’està complint el condition coverage.

*-Path coverage:* Les proves realitzades compleixen tots els paths independents de la funció.

Complexitat ciclomàtica:

M = num. de condicions + 1 = 1 + 1 = 2

O també:

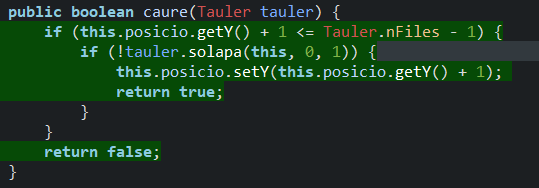
M num.arestes - num.nodes + 2 = 4 - 4 + 2 = 0 + 2 = 2

C:\Users\marcm\AppData\Local\Microsoft\Windows\INetCache\Content.Word\posicioValida.png

1. (1A, 2-No, 4 )

2. (1A, 2-Si, 3-B, 4)

**Funció caure():**



-*Statement coverage:* Totes les sentències s’estan executant com a mínim un cop.

-*Decision coverage:* Totes les decisions prenen els valors true i false.

-*Condition coverage:* Com en aquest codi no tenim cap condició amb més d’una decisió, el fet de que es compleixi el decision coverage ja ens garantitza que s’està complint el condition coverage.

*-Path coverage:* Les proves realitzades compleixen tots els paths independents de la funció.

Complexitat ciclomàtica:

M = num. de condicions + 1 = 2 + 1 = 3

C:\Users\marcm\AppData\Local\Microsoft\Windows\INetCache\Content.Word\posicioValida(1).png

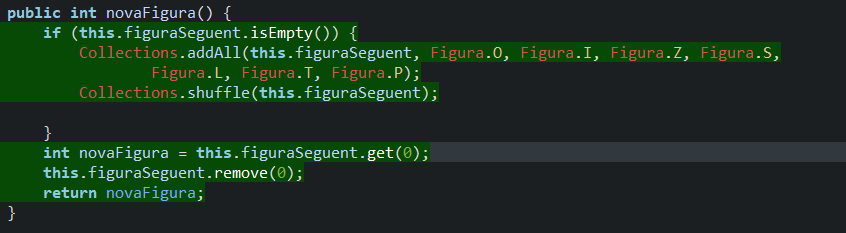
1. (1A, 2-No, 5)

2. (1A, 2-Si, 3-No, 5)

3. (1A, 2-Si, 3-Si, 4-B, 6)

**3. Classe GeneradorFiguraAleatoria**

**Funció novaFigura():**



-*Statement coverage:* Totes les sentències s’estan executant com a mínim un cop.

-*Decision coverage:* Totes les decisions prenen els valors true i false.

-*Condition coverage:* Com en aquest codi no tenim cap condició amb més d’una decisió, el fet de que es compleixi el decision coverage ja ens garantitza que s’està complint el condition coverage.

*-Path coverage:* Les proves realitzades compleixen tots els paths independents de la funció.

Complexitat ciclomàtica:

M = num. de condicions + 1 = 1 + 1 = 2

C:\Users\marcm\AppData\Local\Microsoft\Windows\INetCache\Content.Word\posicioValida(2).png

1. (1A, 2-No, 4-C, 5)

2. (1A, 2-Si, 3-B, 4-C, 5)